B-Spline Method for Energy Minimization in Grid-Based Molecular Mechanics Calculations

DANIEL OBERLIN, JR., HAROLD A. SCHERAGA

Baker Laboratory of Chemistry, Cornell University, Ithaca, New York 14853-1301

Received 19 June 1997; accepted 14 August 1997

ABSTRACT: A method is described for molecular mechanics calculations based on a cubic B-spline approximation of the potential energy. This method is useful when parts of the system are allowed to remain fixed in position so that a potential energy grid can be precalculated and used to approximate the interaction energy between parts of a molecule or between molecules. We adapted and modified the conventional B-spline method to provide an approximation of the Empirical Conformational Energy Program for Peptides (ECEPP) potential energy function. The advantage of the B-spline method over simpler approximations is that the resulting B-spline function is C2 continuous, which allows minimization of the potential energy by any local minimization algorithm. The standard B-spline method provides a good approximation of the electrostatic energy; but in order to reproduce the Lennard–Jones and hydrogen-bonding functional forms accurately, it was necessary to modify the standard B-spline method. This modification of the B-spline method can also be used to improve the accuracy of trilinear interpolation for simulations that do not require continuous derivatives. As an example, we apply the B-spline method to rigid-body docking energy calculations using the ECEPP potential energy function. Energies are calculated for the complex of Phe-Pro-Arg with thrombin. For this system, we compare the performance of the B-spline method to that of the standard pairwise summation in terms of speed, accuracy, and overhead costs for a variety of grid spacings. In our rigid-body docking calculations, the B-spline method provided an accurate approximation of the total energy of the system, and it resulted in an 180-fold reduction in the time required for a single energy and gradient calculation for this system. *J Comput Chem* **19**: 71–85, 1998

Correspondence to: H. A. Scheraga; e-mail: has5@cornell.edu
Grant sponsor: NIH; grant number GM-14312
Grant sponsor: NSF; grant number MCB95-13167

CCC 0192-8651 / 98 / 010071-15

Keywords: molecular docking; protein folding; grid approximation; trilinear interpolation; potential energy

Introduction

In chemical simulations such as protein folding and molecular docking, it is important to be able to calculate and minimize molecular potential energy functions efficiently. Speed becomes critical in the treatment of larger systems because the number of pairwise interactions grows quadratically with the number of atoms in the system. In order to reduce the number of interactions that must be computed, some methods[1–4] make the assumption that the host molecule, or a portion of it, remains rigid (in molecular docking) so that each ligand atom interacts with a static potential field defined by the fixed atoms of the host molecule. It may also be possible to apply these methods to protein folding when parts of a polypeptide chain remain fixed. Hereafter, for illustrative purposes, the discussion will be presented in terms of molecular docking. The potential energy field of the host molecule is computed for points on a 3-dimensional grid before the simulation begins. During the simulation, the double summation of pairwise interactions between ligand and host atoms is replaced by a single sum over ligand atoms, each of which interacts with the field of the host molecule. The potential energy for a single ligand atom can be approximated quickly by using a weighted average of the potential energy at nearby grid points. In previous works[2–4] the grid approximation was based on trilinear interpolation,[5] which uses the eight grid points at the vertices of a cubic grid cell surrounding a ligand atom to approximate the potential energy at the location of that atom. One disadvantage of this method is that it does not provide a continuous derivative across the boundaries of the grid cells, making the resulting function unsuitable for local minimization algorithms such as Newton-type methods. To use a grid method with these algorithms, a functional form is needed that will provide the degree of continuity required by the particular algorithm.

In this work we use a B-spline[6] method to provide a smooth approximation to the potential energy field of the host molecule. A B-spline is a piecewise polynomial function that is used to parameterize smooth curves or surfaces from a set of discrete data (such as a grid of values of potential energy). B-splines can be constructed from polynomials of any degree, and higher degree B-splines produce smoother curves with a higher order of continuity. In the method presented here, use is made of a cubic B-spline that provides a continuous second derivative. The details of the approximation are presented in a later section of this article.

B-splines have been used previously in molecular mechanics calculations for the approximation of Ewald sums.[7] In the present work we adapted and modified the B-spline method to provide a direct approximation to the Empirical Conformational Energy Program for Peptides (ECEPP) potential energy function rather than using it to approximate an Ewald sum. ECEPP includes the Lennard–Jones (12–6), hydrogen-bond (12–10), and electrostatic ($1/\mathbf{r}$) functional forms that are common to many empirical potential energy functions. The resulting B-spline function is C2 continuous, which ensures that it is suitable for minimization by any Newton-type algorithm. The standard B-spline method provides a good approximation of the electrostatic energy; but in order to reproduce the Lennard–Jones and hydrogen-bonding functional forms accurately, it was necessary to modify the standard B-spline method for reasons that will be discussed in a later section. In addition, the modification of the B-spline method can also be used to improve the accuracy of trilinear interpolation for simulations that do not require continuous derivatives.

As an example, we develop and apply the B-spline method to rigid-body docking using the ECEPP potential energy function. Energies are calculated for the complex of Phe-Pro-Arg with thrombin. For this system we compare the performance of the B-spline method to that of the standard pairwise summation in terms of speed, accuracy, and overhead costs for a variety of grid spacings.

Energy Function

The components of the ECEPP/3 potential energy function[8] consist of electrostatic, Lennard–

Jones, hydrogen-bonding, and intrinsic torsional terms. In this rigid-body docking example, we consider only the intermolecular interactions between the ligand and host molecules. The resulting pairwise summation can be written as

$$F = \sum_{i=1}^{N_a} \sum_{j=1}^{N_b} \frac{q_i q_j}{|\mathbf{r_i} - \mathbf{r_j}|} + \frac{A_{hk}}{|\mathbf{r_i} - \mathbf{r_j}|^{12}} + \frac{B_{hk}}{|\mathbf{r_i} - \mathbf{r_j}|^{10}} + \frac{C_{hk}}{|\mathbf{r_i} - \mathbf{r_j}|^{6}}, \quad (1)$$

where h is the type of atom i of the host molecule and k is the type of atom j of the ligand molecule. The parameters A, B, and C are determined by the interaction type. Either B is zero (for the Lennard–Jones terms) or C is zero (for the hydrogen-bonding terms). N_a and N_b are the number of host and ligand atoms, respectively. Because the positions of the host atoms in the inner sum are fixed, we sum over these atoms and define the potential energy for a *single* ligand atom with respect to all of the fixed atoms of the host:

$$\Phi_{\mathrm{el}}(\mathbf{r}_j) = \sum_{i=1}^{N_a} \frac{q_i}{|\mathbf{r_i} - \mathbf{r_j}|},$$

$$\Phi_k(\mathbf{r}_j) = \sum_{i=1}^{N_a} \frac{A_{hk}}{|\mathbf{r_i} - \mathbf{r_j}|^{12}} + \frac{B_{hk}}{|\mathbf{r_i} - \mathbf{r_j}|^{10}} + \frac{C_{hk}}{|\mathbf{r_i} - \mathbf{r_j}|^{6}}, \quad (2)$$

where Φ_{el} represents the electrostatic potential energy and Φ_k represents the Lennard–Jones and hydrogen-bond potential energy for a ligand atom of type k. With these functions defined, we can rewrite the total interaction energy between the host and the ligand as a single sum over all of the ligand atoms:

$$F = \sum_{j=1}^{N_b} q_j \Phi_{\mathrm{el}}(\mathbf{r_j}) + \Phi_k(\mathbf{r_j}). \quad (3)$$

When applying the B-spline method to simulations of flexible molecules, the standard ECEPP torsional terms are calculated in addition to the terms approximated by the B-spline.

In the method presented here, we define a box that contains the region of the host molecule where ligand docking will occur. The potential energy functions Φ are computed at points defined by a 3-dimensional grid bounded by this box. Each grid of potential energy values is computed once before the simulation begins. During the simulation, the values and derivatives of Φ may be approximated at any point inside the box by combining the values of the potential energy at nearby grid points using the B-spline method. For notation we will designate a grid G of potential energy values as

$$G_{lmn} = \Phi(\mathbf{r}_{lmn}), \quad (4)$$

where l, m, and n are the discrete indices over the space of the grid. For the ECEPP/3 potential energy, a single grid is computed for the electrostatic potential energy, and N_{type} noncovalent grids (one for each type of atom in the ligand) are computed for the Lennard–Jones and hydrogen-bonding potential energies. The electrostatic grid represents the electrostatic potential energy of the host molecule, and each noncovalent grid represents the Lennard–Jones and hydrogen-bonding potential energies for a particular atom type with respect to all atoms of the host molecule. Each grid point has potential energy contributions from *all* atoms of the host; it is intended to be used for a *single* type of ligand atom. During the course of computing the noncovalent grids for the different atom types, it is necessary to compute the inverse powers of **r** (the distance between a grid point and a fixed atom in the host) for every pairing of a grid point and a host atom. When computing this set of potential energy grids, it is especially important to implement an efficient algorithm that does not waste time by repeatedly calculating the same inverse powers of **r** for each grid. In Appendix A, we present an algorithm that resulted in a nine-fold speedup in the time required to generate the 13 grids needed for the ECEPP function.

B-Spline Approximation

The B-spline methodology was originally developed for use in the computer graphics industry, where it is necessary to use functions that parameterize smooth curves and surfaces from an input of discrete points. In this work, we use B-splines to provide a smooth approximation of the potential energy field of the host molecule in rigid-body docking energy calculations. Before the simulation begins, the potential energy Φ is computed for points that lie on a 3-dimensional grid G encompassing the volume of interest of the host molecule. During the simulation, the spline is used to approximate the potential energy at the locations of the ligand atoms by combining the potential energy values that were computed previously at nearby grid points.

To introduce the cubic B-spline grid approximation described in this article, we first consider a linear B spline, which is equivalent to the method of trilinear interpolation.[5] In this method, a cubic cell is defined by locating the eight grid points surrounding the point P in space where the potential energy of interaction of the host with a ligand atom is to be evaluated (see Fig. 1). The potential energy values that were computed previously at these grid points are weighted and summed to approximate the potential energy of the host at the point P inside the cube. We will refer to the vertex of the cube with the smallest indices l, m, and n as the ''origin vertex'' of the cell, and will designate the origin vertex as

$$\mathbf{o} = \mathbf{r}_{lmn}. \tag{5}$$

The approximate potential energy at a given point P inside the cube is obtained by forming a linear combination of the potential energy values at the eight surrounding grid points, each weighted by a product of three 1-dimensional basis functions:

$$\Phi(\mathbf{r}) \approx \sum_{i=l}^{l+1} \sum_{j=m}^{m+1} \sum_{k=n}^{n+1} G_{ijk} B^1_{i-l}(\mathbf{r}_x - \mathbf{o}_x) \times B^1_{j-m}(\mathbf{r}_y - \mathbf{o}_y) B^1_{k-n}(\mathbf{r}_z - \mathbf{o}_z), \tag{6}$$

where $\mathbf{r}$ is the location of point P where the potential energy is being evaluated, G_{ijk} are the values of the potential energies Φ that were computed previously, and B^1 refers to the set of two linear B-spline basis functions that are defined by[6]

$$B^1_0(x) = 1 - \frac{x}{d},$$
$$B^1_1(x) = \frac{x}{d}. \tag{7}$$

In eq. (7), d is the spacing between two adjacent grid points. The basis functions give a higher weight to grid points that are closer to the point P where the potential energy is being approximated. The basis functions also have the property known as partition of unity:

$$\sum_{i=0}^{1} \sum_{j=0}^{1} \sum_{k=0}^{1} B^1_i(x) B^1_j(y) B^1_k(z) = 1. \tag{8}$$

In the special case of linear B-splines, the spline interpolates the potential energy values at the grid points. That is, when the spline is evaluated at the exact location of a grid point, that grid point is given a weight of one and all other grid points are given a weight of zero. This results in perfect agreement between the spline and potential energy function at the grid points. This method of interpolation produces a piecewise linear function of the form of eq. (7). As a result, the function has discontinuous first derivatives at the grid cell boundaries, thus making it unsuitable for most local minimization algorithms. A remedy to this problem is to use B-splines of higher degree. Higher degree B-splines produce piecewise polynomial surfaces with a higher degree of continuity, but it is necessary to combine more potential energy values to produce the approximation. Cubic B-splines have the property that their first derivatives are smooth, making them ideal for local minimization.

For a 3-dimensional B-spline of degree p, the potential energy values at $(p + 1)^3$ grid points must be used to construct the approximation. In the case of a linear B-spline where $p = 1$, this produces the eight grid points that are the vertices of the cell containing the location where the potential energy is approximated (as in Fig. 1). For a cubic spline where $p = 3$, the 64 points in a $4 \times 4 \times 4$ lattice are used to produce the approximation of the potential at any point P in the *central* cell of this lattice (see Fig. 2).

The spline approximation is obtained by a linear combination of the potential energy values G_{ijk} at the grid points comprising the $4 \times 4 \times 4$ lattice,

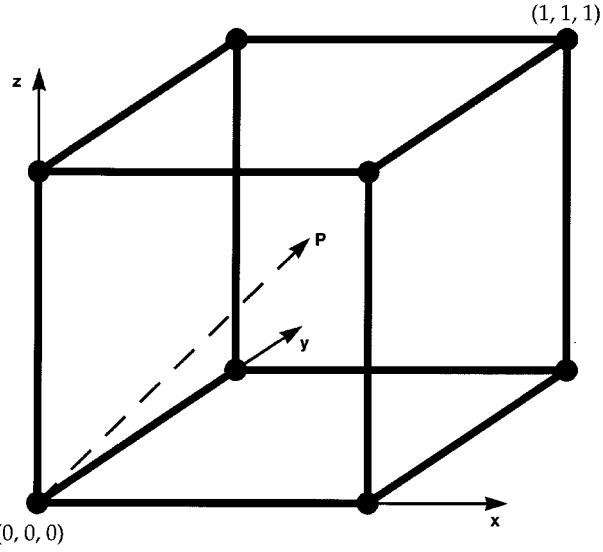

FIGURE 1. The $2 \times 2 \times 2$ cubic grid cell for which the potential energies of the host at the eight vertices are used for trilinear interpolation of the potential energy at a point P within the cell where a ligand atom is located.

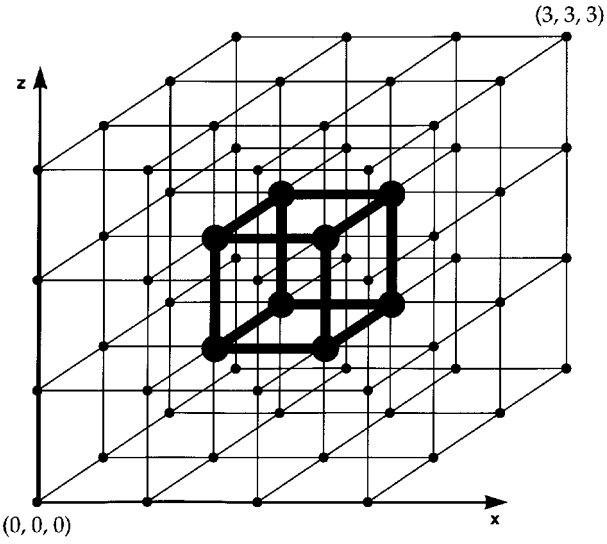

FIGURE 2. The 4 × 4 × 4 lattice of points used for the cubic B-spline approximation. The potential energy at any point *P* in the central cell of the lattice is evaluated in terms of the values at all the points of the 4 × 4 × 4 lattice.

each weighted by a product of three cubic basis functions. The cubic B-spline approximation becomes

$$\Phi(\mathbf{r}) \approx \sum_{i=l}^{l+3} \sum_{j=m}^{m+3} \sum_{k=n}^{n+3} G_{ijk} B^3_{i-l}(\mathbf{r}_x) B^3_{j-m}(\mathbf{r}_y) B^3_{k-n}(\mathbf{r}_z), \tag{9}$$

where B^3 refers to the set of four cubic B-spline basis functions that are distinguished by the four possible values of the subscript for B. The cubic B-spline basis functions (as well as the basis functions for any degree B spline) are defined by a recursive formula presented in Appendix B. A similar recursive formula exists for the derivatives of the basis functions, so that the B-spline summation may be differentiated to provide the function gradient.

It is important to note that B-splines of degree greater than 1 will approximate the potential energy values at the grid points without necessarily passing through them. That is, the approximation will not match the potential energy values at the grid points exactly. In practice, the B-spline functional form provides excellent accuracy in the approximation of the electrostatic potential function $1/\mathbf{r}$, but in order to obtain a satisfactory accuracy for the Lennard–Jones and hydrogen-bonding potential energy functions it is necessary to modify the method.

The surprising accuracy for $1/\mathbf{r}$ is probably due to the fact that it is a harmonic function of the variables $\mathbf{r}_x$, $\mathbf{r}_y$, and $\mathbf{r}_z$. An interesting property of harmonic functions $f(\mathbf{x})$ is that they satisfy the following equation[9]:

$$f(\mathbf{x}) = \frac{1}{A} \cdot \int_{|\mathbf{y}-\mathbf{x}|=r} f(\mathbf{y})\, dS, \tag{10}$$

where A is a constant equal to the surface area of the sphere of integration. In other words, a harmonic function has the special property that the function value at a given point is equal to the average of the function over the surface of a sphere centered at that point. The function evaluation in eq. (10) is similar to the B-spline procedure in the sense that a function value is determined by an average over the function values at nearby points. In the case of the B-spline procedure, a finite number of nearby grid points are averaged together. If we consider the B-spline procedure to be an approximation of eq. (10), then we would expect the B-spline to give a better approximation for harmonic functions because eq. (10) holds for these functions. As will be shown in the Results section, this appears to be the case.

The Lennard–Jones and hydrogen-bonding potential functions are not harmonic. Therefore, as shown in the next section, they should be modified in order to apply a B-spline approximation.

Empirical Error Correction

In the previous section, it was shown that the B-spline approximation of the pairwise sum is formed by a linear combination of potential energy values computed at points that lie on a 3-dimensional grid [eq. (9)]. The conventional pairwise sum [eq. (1)] is replaced by a single sum over the ligand atoms [eq. (3)], each of which interacts with the field of the entire host molecule. In order to approximate the potential energy of the host at the location of a ligand atom, the B-spline procedure is used to combine the potential energies at neighboring grid points [eq. (9)]. Each of these grid potential energy values is computed prior to the simulation by a summation of the distance dependent interactions between an atom placed at the grid point and the set of fixed atoms belonging to the host molecule [eqs. (2), (4)].

In a discussion of the sources of error in the approximation, we will refer to the error in the

approximation of a single interaction term. This is the error that results from the spline approximation of an interaction term ($\mathbf{r}^{-1}$, $\mathbf{r}^{-6}$, $\mathbf{r}^{-10}$, or $\mathbf{r}^{-12}$) between a ligand atom and a single fixed atom in the host molecule. In this example of a single interaction term, each of the grid points would represent the interaction potential energy of a single host atom rather than the sum of the potential energies of each of the host atoms. Although we are interested in the error in the sum of potential energies, the error in the sum is directly related to the error in a single term. Because the B-spline is a linear transformation of the sum of potential energy values, the error in the approximation of the sum will be equal to the sum of the errors in the approximations of each of the individual terms; thus, the error in the approximation should be analyzed in terms of the error in the individual terms.

Another important consideration is that the B-spline of one of these pairwise interaction terms will not exhibit exact radial symmetry with respect to the position of the fixed host atom. This is because the approximation is composed of contributions from potential energy values that have been sampled at points lying on a cubic grid. Thus, the exact value for a given distance $\mathbf{r}$ will also depend on the locations of the host and ligand atoms within the grid. For this reason, when we examine the accuracy of the B-spline of a single interaction for a given interatomic distance $\mathbf{r}$, we must consider the mean value of the spline over all possible pairs of points corresponding to a host and ligand atom separated by a distance $\mathbf{r}$, which we denote as $\langle f_{\text{spline}} \rangle_r$.

For inverse powers of $\mathbf{r}$ other than $\mathbf{r}^{-1}$, where the standard B-spline method results in a large degree of error, we observed that this error is due to a ''shifting effect'' that is a result of the B-spline transformation. The mean value of the spline approximation is shifted to lower values of $\mathbf{r}$ with respect to the correct value (i.e., the graph is shifted to the right):

$$\langle f_{\text{spline}} \rangle_r \approx f(r - s(r)). \tag{11}$$

Here, the function $s(r)$ represents the degree to which the spline function has been shifted toward smaller values of r. The function s varies for different grid spacings as well as for different functions f. As an example, for the Lennard–Jones function each of the two terms (attractive and repulsive) will be shifted by a different function $s(r)$. The cumulative effect of these errors in the pairwise terms results in a large error in the total energy.

The systematic error described in eq. (11) can be corrected empirically by replacing the function f with a modified function f_c. In this approach, we compensate for the right shift of the mean value of the B-spline to lower values of $\mathbf{r}$ by shifting the original function f to higher values (i.e., to the left of the graph).

$$f_c(r) = f(r + s(r)). \tag{12}$$

The spline is then calculated using the values of the function f_c rather than f, which results in a substantial reduction in the error. For the terms required for the Lennard–Jones and hydrogen-bonding functions, we found that the following functional form is useful in describing the shifting function $s(r)$ over an acceptable range of r:

$$s(r) \approx a + \frac{b}{c + r}, \tag{13}$$

where the parameters a, b, and c were determined by a fitting procedure that minimizes the error in $\langle f_{\text{spline}} \rangle_r$. In this fitting procedure, the following estimation of the error in the spline approximation is minimized for the set of interatomic distances r_i:

$$E(a, b, c) = \sum_i \left(\frac{\langle f_c \rangle_{r_i} - f(r_i)}{f(r_i)} \right)^2, \tag{14}$$

where f_c is given by eq. (12) and $f(r_i)$ is a term of the ECEPP pairwise interaction functions: either r^{-6}, r^{-10}, or r^{-12}.

We found that a good set of parameters could be determined by choosing a representative set of values of r_i (viz., 1.75, 3.0, and 6.0 Å) where the function is most sensitive to the magnitudes of the distances. The average value of the spline function $\langle f_c \rangle_{r_i}$ was computed over a sample of 1000 randomly generated pairs of points that were contained within the grid and were separated by the distance r_i. A single local minimization was performed starting with each parameter equal to 1, and the Broyden–Fletcher–Goldfarb–Shanno (BFGS) quasi-Newton algorithm[10] was used to perform the local minimization. Because of the dependence of the function $s(r)$ on the particular function f and the grid spacings, it was necessary to repeat this procedure at a variety of grid spacings for each of the three terms r^{-6}, r^{-10}, and r^{-12}.

In summary, in order to compensate for the right-shifting effect of the B-spline, the functions r^{-6}, r^{-10}, and r^{-12} are replaced by left-shifted functions that have been parameterized so that the error of the B-spline approximation of these functions is minimized. The summations of these shifted functions are computed for the grid points in the same manner as the original function [eqs. (2), (4)], and the spline approximation is used to combine the values of the shifted functions. Because the B-spline is a linear transformation, the correction of the individual pairwise terms results in the proper correction of the total energy function.

Results

To evaluate the properties of the modified B-spline method, we present results from energy calculations for a rigid-body docking system consisting of the protein thrombin as the host molecule and the tripeptide Phe-Pro-Arg as the ligand molecule. In this system the host molecule is stationary and the ligand molecule is allowed full translational and rotational freedom. Both molecules are rigid and are not allowed to change conformation. The six variables of the system are the three translational and three rotational degrees of freedom for the ligand. The interaction energy is formulated as the summation of the interaction energies of the ligand atoms with respect to the field of the host molecule as in eq. (3), and each value for Φ is approximated using the cubic B-spline formulation in eq. (9). The potential energy values Φ_{ijk} on the grid are calculated using the ECEPP potential energy parameters and the correction procedure described in the previous section. The structures of the host and ligand were obtained from the crystal structure lppb.pdb of the thrombin inhibitor Phe-Pro-Arg-chloromethylketone (PPACK) bound to thrombin.[11] The chloromethylketone group of PPACK was removed in order to simulate the formation of a noncovalent complex. The resulting structure of the tripeptide Phe-Pro-Arg was then regularized to the ECEPP geometry, and the energy of the binary complex was minimized to produce the structures used in these calculations.

Before the modified B-spline method could be applied, it was necessary to obtain the parameters a, b, and c of the shift function $s(r)$ [eq. (13)] for the terms r^{-6}, r^{-10}, and r^{-12} of the pairwise interaction potentials. The parameters were found by minimizing the function defined in eq. (14). These parameters were determined for a variety of grid spacings, and Table I lists the resulting parameters. For testing purposes, three grids of $50 \times 50 \times 50$ Å dimensions were calculated for a selection of the grid spacings in Table I, namely 0.25,

TABLE I.
Cubic B-Spline Shift Function Parameters of Eq. (13) Determined for Different Grid Spacings.

Grid Spacing (Å)	Function	a	b	c
1.0	r^{-6}	-6.3323×10^{-2}	1.6120	2.8975
	r^{-10}	-3.6372×10^{-1}	7.4107	6.8799
	r^{-12}	-9.2671×10^{-1}	2.2550×10^{1}	1.3136×10^{1}
0.8	r^{-6}	-1.8287×10^{-2}	7.4426×10^{-1}	1.4382
	r^{-10}	-6.0632×10^{-2}	1.7326	2.3502
	r^{-12}	-9.0247×10^{-2}	2.3743	2.8494
0.66	r^{-6}	-8.0604×10^{-3}	4.6981×10^{-1}	9.5659×10^{-1}
	r^{-10}	-2.5977×10^{-2}	9.9251×10^{-1}	1.5294
	r^{-12}	-3.7353×10^{-2}	1.2948	1.8137
0.5	r^{-6}	-3.4953×10^{-3}	2.4732×10^{-1}	5.5514×10^{-1}
	r^{-10}	-9.6628×10^{-3}	4.8727×10^{-1}	8.9241×10^{-1}
	r^{-12}	-1.3664×10^{-2}	6.2053×10^{-1}	1.0611
0.33	r^{-6}	-6.8142×10^{-4}	1.0000×10^{-1}	2.4346×10^{-1}
	r^{-10}	-1.9936×10^{-3}	1.8873×10^{-1}	4.0201×10^{-1}
	r^{-12}	-2.9069×10^{-3}	2.3620×10^{-1}	4.8497×10^{-1}
0.25	r^{-6}	-2.1119×10^{-4}	5.4371×10^{-2}	1.3572×10^{-1}
	r^{-10}	-6.2407×10^{-4}	1.0058×10^{-1}	2.2551×10^{-1}
	r^{-12}	-9.1829×10^{-4}	1.2470×10^{-1}	2.7334×10^{-1}

0.5, and 1.0 Å. The results of these tests are shown below.

The energy of the rigid-body docking system (approximated by the B-spline method) was then minimized starting from 500 random configurations in order to confirm that the B-spline function is minimizable and to provide an ensemble of low-energy structures with which to evaluate the accuracy of the approximation. The minimizations were performed using the BFGS algorithm, and the 0.5 Å grid was used for the energy minimizations. Later, the 0.25 and 1.0 Å grids were used to evaluate the accuracies of these grid spacings for the minima generated by this procedure.

Because the starting configurations were generated randomly, a significant number of the resulting minima had severe atomic overlaps that resulted in extremely high energy values. In these configurations, the B-spline method tended to significantly underestimate the true value of the energy. Figure 3 is a plot of the log of the B-spline approximation versus the log of the exact energy values for the 291 local minima that had positive energies out of the total of 500 local minima. The wide range in magnitudes of the energy values is due to the steepness of the repulsive r^{-12} term for the small values of r resulting from the overlapping atoms. Because the B-spline approximation is produced from a weighted sum of the potentials at the 64 grid points surrounding the ligand atom, the maximum value of the spline inside a grid cell is limited by the highest of the 64 values in the sum.

Of the 500 local minima that were generated, we selected the 209 minima with negative energy values to represent structures without significant atomic overlaps. For these low-energy configurations, Figures 4–6 are plots of the B-spline energies of these minima for the 1.0, 0.5, and 0.25 Å grids, respectively, versus the exact energies. Table II lists the root mean square (RMS) errors for these calculations. The tables and figures show that the error of the approximation is reduced for smaller grid spacings, as would be expected.

Figure 7 is a plot of the B-spline total energies for an uncorrected 0.5 Å grid versus the exact energies for the low-energy configurations. The RMS errors for these calculations are listed in Table II with the errors for the corrected grids. This figure illustrates the significant error associ-

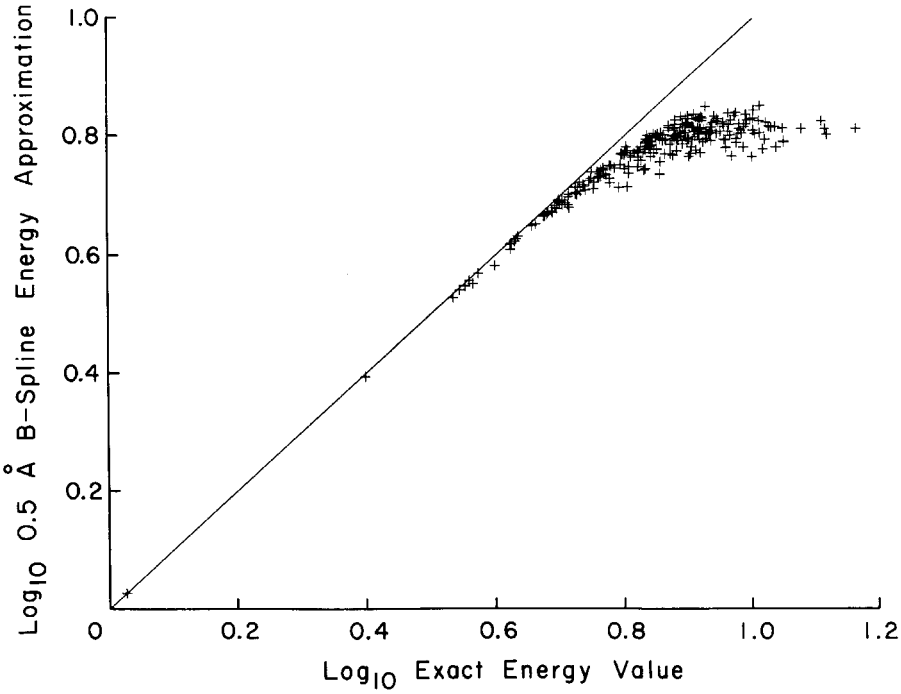

FIGURE 3. Logarithmic plot of the corrected B-spline energy values versus the exact energy values for the 291 local minima that had positive energy values. A 0.5 Å grid was used for these B-spline calculations. Each + symbol represents the B-spline energies and exact energies for a single local minimum. The line $y = x$ illustrates the deviation of the B-spline approximation from the exact values.

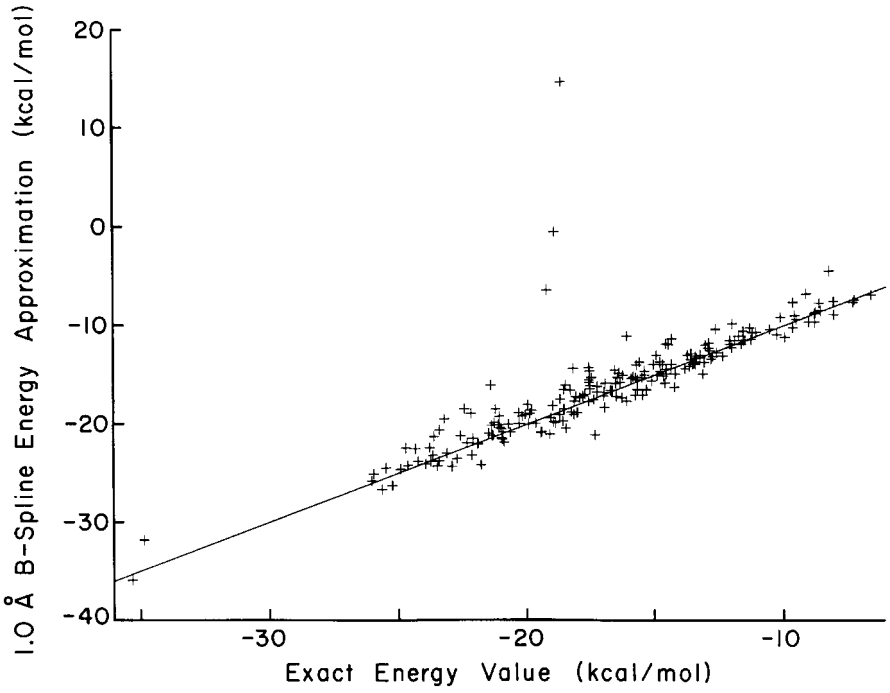

FIGURE 4. Plot of the corrected B-spline energy values versus the exact energy values for the 209 local minima that had negative energy values. The plot is for a grid with a spacing of 1.0 Å.

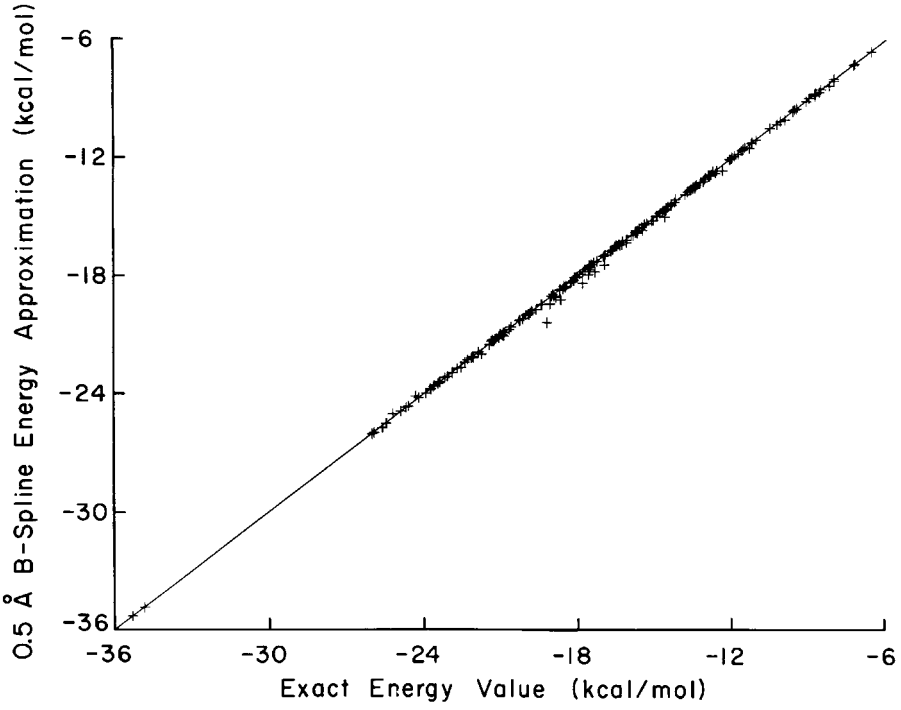

FIGURE 5. Same as Figure 4, but for a grid spacing of 0.5 Å.

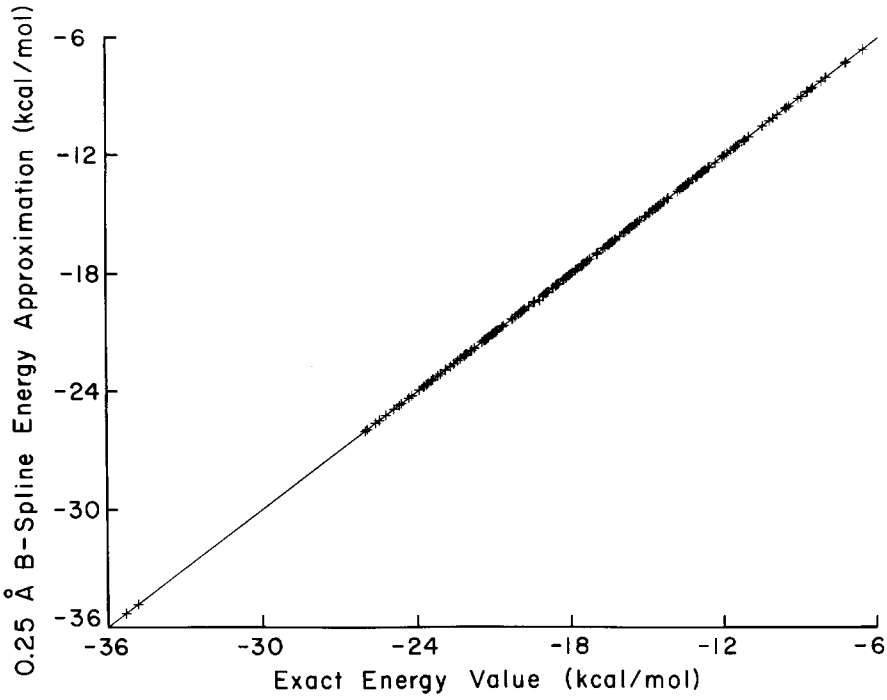

FIGURE 6. Same as Figure 4, but for a grid spacing of 0.25 Å.

ated with the shifting effect of the standard B-spline method. The B-spline energies tend to have large variations in magnitude and provide a very poor approximation to the exact energy. Figure 8 is a plot of the electrostatic energy of the uncorrected B-spline approximation versus the exact electrostatic energy. Table II also lists the RMS error for the electrostatic energy. These data show the remarkable accuracy of the B-spline in the approximation of the function r^{-1} and illustrate that the error in the total energy is due entirely to the shifting effect of the B-spline on the pairwise energy terms r^{-6}, r^{-10}, and r^{-12}.

TABLE II.
RMS Errors of Low-Energy Minima for Various Grid Calculations.

Grid Spacing (Å)	Type of Calculation	RMS Error of Energies (kcal / mol)
1.0	Corrected B spline	3.1
0.5	Corrected B spline	0.13
0.25	Corrected B spline	0.012
0.5	Uncorrected B spline	21.8
0.5	Electrostatic B spline	0.00043
0.5	Uncorrected trilinear	5.53
0.5	Corrected trilinear	0.39

The shifting effect is not unique to cubic B-splines, and it is also observed in trilinear interpolation. In studies in which continuity of the derivatives is not necessary (such as Monte Carlo calculations), the correction method can also be used to improve the accuracy of the trilinear interpolation method. As a demonstration, we calculated the energies and errors of the 209 low-energy configurations using the trilinear approximation [eqs. (6), (7)] and the same uncorrected 0.5 Å grid that was used for the B-spline calculations in Figure 7. We then applied the fitting technique to find the parameters a, b, and c for the shifting functions that would minimize the error for a trilinear interpolation using a 0.5 Å grid. The resulting parameters are listed in Table III, and the RMS error for the uncorrected and corrected energies are listed in Table II. These data show that the error of the *uncorrected* trilinear approximation is significantly less than that of the *uncorrected* B-spline approximation. This is due to the fact that the trilinear approximation interpolates the function values and is thus more tightly constrained to these values than the cubic B-spline. Higher order B-splines involve averaging over more function values, and they tend to produce smoother curves at the expense of larger deviations from the defining function values. Table II also shows that the

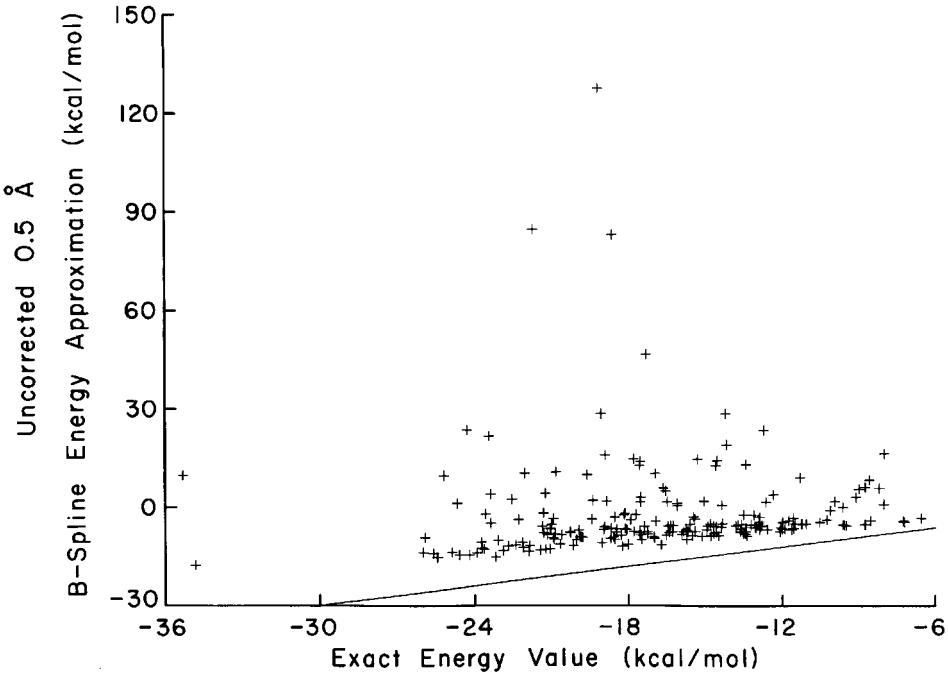

FIGURE 7. Plot of the uncorrected B-spline energy values versus the exact energy values for the 209 local minima that had negative energy values. A 0.5 Å grid was used for these B-spline calculations.

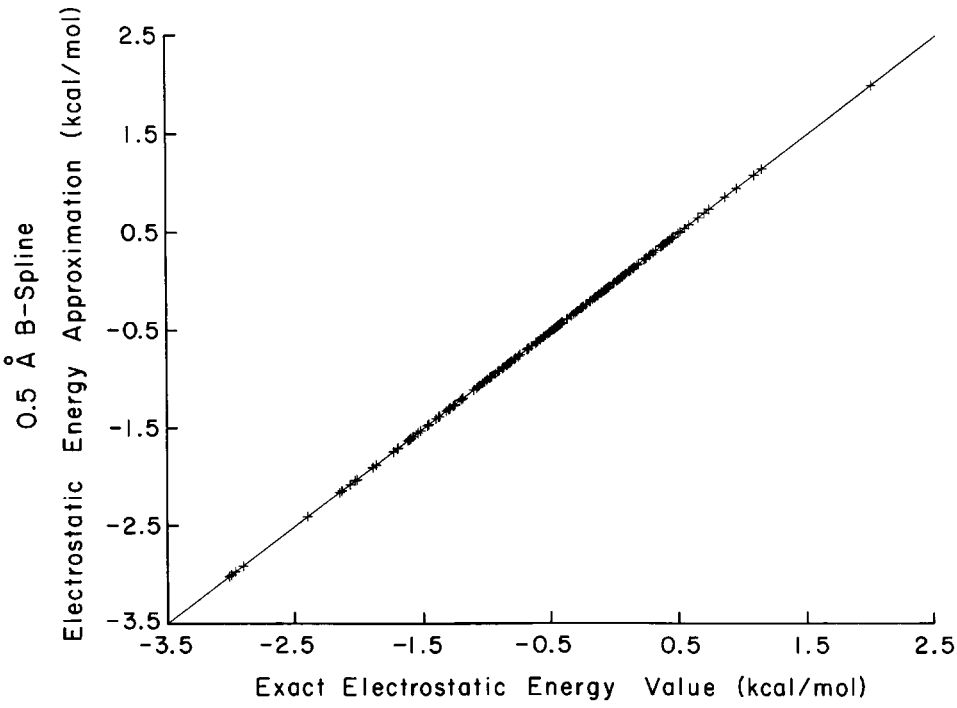

FIGURE 8. Plot of the electrostatic B-spline energy values versus the exact energy values for the 209 local minima that had negative energy values. A 0.5 Å grid was used for these B-spline calculations.

TABLE III.
Trilinear Interpolation Shift Function Parameters of Eq. (13).

Grid Spacing (Å)	Function	*a*	*b*	*c*
0.5	r^{-6}	-1.9439×10^{-4}	1.0906×10^{-1}	2.0767×10^{-1}
	r^{-10}	-1.7222×10^{-3}	2.1087×10^{-1}	4.3926×10^{-1}
	r^{-12}	-2.9061×10^{-3}	2.6693×10^{-1}	5.6196×10^{-1}

error of the *corrected* B-spline approximation is less than the error of the *corrected* trilinear approximation, illustrating that the function can be represented better by a cubic polynomial than by a linear function.

In the rigid-body docking calculations presented here, the B-spline method resulted in a 180-fold reduction in the time required for a single energy and gradient calculation for this system as compared to the standard pairwise calculation. The timings required to calculate the 13 grids (corresponding to the 13 unique types of atoms in ECEPP) needed to perform ECEPP calculations are listed in Table IV. These calculations were performed using the second algorithm described in Appendix A. This algorithm was 9 times faster (with 13 atom types) than the simpler conventional method listed first in Appendix A. These timings were made on a single processor of an SGI Power Onyx system. The storage required for grids of various sizes are also listed in Table IV.

Conclusions

The modified B-spline method provides a fast and accurate approximation of standard molecular mechanics force field functions for applications in which part of the system can be considered fixed. Because the approximation has a continuous second derivative, the resulting function is minimizable by gradient-based local minimization algorithms. Thus, simulations that depend on such minimization algorithms can benefit from the B-spline method. If the simulation does not require a continuous derivative (as in the case of Monte Carlo calculations), the B-spline correction procedure can also be used to provide an accurate trilinear approximation.

TABLE IV.
Grid Storage Requirements and Timings for Grid Generation.

Grid Spacing (Å)	Generation Time (min)	Storage Required (MB)
1.0	16	3.5
0.5	137	28.0
0.25	1105	224.0

In this example, we demonstrated the use of the method for rigid-body docking calculations. An important point about the timing of the method is that, for rigid-body docking, the time for the energy calculation scales linearly with the number of atoms in the ligand. It is also possible to apply the method to situations in which the ligand is completely flexible or when only part of the host molecule is fixed. In other work,[12] we obtained a 45-fold speed improvement in simulations of the same system but used a completely flexible ligand (compared with the 180-fold improvement for rigid-body docking).

The spline correction procedure is simple to implement and requires no additional complexity in the actual energy evaluation procedure. It is interesting to note that, without the correction procedure, the B-spline and trilinear approximations (for terms other than r^{-1}) are in error to a degree that they are essentially useless.

Although the accuracy of the method is very good for low-energy configurations, there is a large underestimation of the high energies of configurations with severe atomic overlaps. This error for high-energy configurations does not limit the usefulness of the method because these high-energy values are not accurately defined by the force field and are generally not of quantitative interest.

Finally, consideration must be given to the overhead and storage costs associated with the generation of these potential energy grids. In this work, the grid spacings that we tested spanned a wide range of accuracy and storage requirements. A 1.0 Å grid is sparse enough so that it can contain a very large system, but it has an associated error that renders it unusable for quantitative work. At

the other extreme, a 0.25 Å grid provides very good accuracy but requires an extremely large amount of storage (and computation time) for a system like the example in this article. It has been our experience that a grid spacing of 0.66 Å provides a good compromise between accuracy and required storage for systems of the size considered in this work. The resulting data have been of manageable size and the approximation has provided a satisfactory accuracy.

Appendix A: Efficient Generation of Potential Energy Grids

When generating the N_{type} interaction grids for the Lennard–Jones and hydrogen-bonding energies, it is of interest to calculate one grid G^k for each atom type k of the ligand such that

$$G^k_{lmn} = \sum_{i=1}^{N_a} \frac{A_{hk}}{|\mathbf{r_{lmn}} - \mathbf{r_i}|^{12}} + \frac{B_{hk}}{|\mathbf{r_{lmn}} - \mathbf{r_i}|^{10}} + \frac{C_{hk}}{|\mathbf{r_{lmn}} - \mathbf{r_i}|^{6}}. \quad \text{(A.1)}$$

The simplest method, which involves generating the grids sequentially, is very inefficient because there is an outer loop over atom types, and the grid for atom type k is generated completely before the grid for atom type $k + 1$ is generated. Inside the loop over atom types, there is a loop over all the atoms in the host molecule. For each atom in the host molecule there is a loop over the grid points, and a contribution for that atom is added to the potential energy at each point in the grid. Briefly, this algorithm is

Outer loop over atom type k
  Fill the grid G^k with zeros
  Loop over atom j of the host to be placed on the grid
    Loop over grid space l, m, n

$$G^k_{lmn} \Leftarrow G^k_{lmn} + \frac{A_{hk}}{|\mathbf{r_{lmn}} - \mathbf{r_i}|^{12}} + \frac{B_{hk}}{|\mathbf{r_{lmn}} - \mathbf{r_i}|^{10}} + \frac{C_{hk}}{|\mathbf{r_{lmn}} - \mathbf{r_i}|^{6}}$$

    End loop over grid space
  End loop over atom i
End loop over atom type k

This method is very wasteful because it calculates the functions r^{-6}, r^{-10}, and r^{-12} a total of N_{type} times for every pairing of a host atom to a grid point. A more efficient method can be used that calculates each of these functions only once for each pairing. During the generation of the grids, the method requires additional temporary storage for three potential energy grids denoted T^{12}, T^{10}, and T^6. This temporary storage is required only during the generation of the grids. The new algorithm can be written as

Create a list L of host atoms by type: $L^k[\,j]$ is the jth atom of type k
Fill the grids G with zeros
Outer loop over atom type k
  Fill the temporary grids T^{12}, T^{10}, and T^6 with zeros
  Loop over atoms of type k, $L^k[\,j]$
    Loop over grid space l, m, n

$$T^{12}_{lmn} \Leftarrow T^{12}_{lmn} + \frac{1}{|\mathbf{r_{lmn}} - \mathbf{r_{L^k[j]}}|^{12}}$$

$$T^{10}_{lmn} \Leftarrow T^{10}_{lmn} + \frac{1}{|\mathbf{r_{lmn}} - \mathbf{r_{L^k[j]}}|^{10}}$$

$$T^{6}_{lmn} \Leftarrow T^{6}_{lmn} + \frac{1}{|\mathbf{r_{lmn}} - \mathbf{r_{L^k[j]}}|^{6}}$$

    End loop over grid space
  End loop over atoms
  Loop over atom types h
    Look up ECEPP parameters A, B, and C for atom types h and k
    Loop over grid space l, m, n

$$G^k_{lmn} \Leftarrow G^k_{lmn} + A_{hk}T^{12}_{lmn} + B_{hk}T^{10}_{lmn} + C_{hk}T^{6}_{lmn}$$

    End loop over grid space
  End loop over atom types
End outer loop over atom types

The second loop over atom types, which updates the final grid G^k, is quite fast compared to the loop over atoms because there are many more atoms than atom types. In ECEPP calculations there are 13 unique atom types, and we observed a ninefold speedup in the time to produce the grids.

Appendix B: Definition of B-Spline Basis Functions

A B-spline is a piecewise polynomial function that produces a smooth parameterization of a sur-

face or function based on values that have been sampled at a set of points defined by a grid. The grid itself represents the parameter space, and the spacing of the grid points along a particular coordinate specifies the values of the corresponding parameter for which the function or surface will be sampled. In the general case, the spacing does not need to be regular; however, for simplicity we chose to use a regular grid spacing in this application. Each grid point is associated with a corresponding value that is either a function value or a vector, depending on whether the spline is being used to parameterize a function or a surface. For the application presented in this article, we parameterized a 3-dimensional function (the potential energy of the host molecule) and the grid has 3-dimensions corresponding to the x, y, and z coordinates of the space in which the potential energy has been sampled. In this section, we introduce the B-spline methodology as applied to 1-dimensional functions and discuss its extension to multidimensional functions.

For the B-spline $S(x)$ of a 1-dimensional function $f(y)$, there is a single parameter x. By analogy with eq. (4) of the text, we will refer to the sampled function values as $G_i = f(y_i)$. The values G_i are associated with the values of the parameter that we denote as x_i. For a parameterized function, the parameter x can be equivalent to the independent variable y, so that $x_i = y_i$. The values x_i are referred to as ''knots,'' and we define a set of numbers known as a ''knot vector'' that lists these values in increasing order.[6] The spline is a piecewise polynomial function, and the knots are boundaries that join the different polynomial functions of which the spline is composed. The knot vector serves the additional purpose of specifying the continuity of the spline at each of the knots. The continuity at a knot may be decreased by repeating the knot within the knot vector. As an example, a knot vector for a cubic B-spline ($p = 3$) with regularly sampled points corresponding to the integers from 1 to 5 would be defined as

$$X = \{1, 1, 1, 1, 2, 3, 4, 5, 5, 5, 5\}. \qquad \text{(B.1)}$$

For a spline of degree p, the degree of continuity at unrepeated knots will be $p - 1$; thus, the interior knots (2–4) will have a continuous second derivative. For each repetition of a knot in the knot vector, the degree of continuity at that knot is decreased by one. For both end points (1, 5), the knots were repeated 3 times, which makes the function itself discontinuous at these points. (This is necessary because the spline ends at these points.) The B-spline methodology was developed so that the continuity at the interior points could also be lowered so that a wide variety of curves and surfaces could be modeled. In this application, the purpose of the spline is to provide a smooth and continuous parameterization of the potential energy function; hence, we do not lower the continuity at any of the interior points.

The polynomial form of the spline is given by the B-spline basis functions. These basis functions are defined in terms of the degree of the spline and the knot vector. From a given knot vector **X**, the B-spline basis functions may be defined with the following recursive formula:[6]

$$B_i^0(x) = \begin{cases} 1 & \text{if } x_i \le x \le x_{i+1} \\ 0 & \text{otherwise} \end{cases},$$

$$B_i^p(x) = \frac{x - x_i}{x_{i+p} - x_i} B_i^{p-1}(x) + \frac{x_{i+p+1} - x}{x_{i+p+1} - x_{i+1}} B_{i+1}^{p-1}(x), \qquad \text{(B.2)}$$

where x_i represent the elements of the knot vector **X**. The B-spline basis functions are used to smoothly combine the function values at nearby knots. A 1-dimensional B-spline of degree p can be written as

$$S(x) = \sum_{i=0}^{n} B_i^p(x) \cdot G_i, \qquad \text{(B.3)}$$

where p is the number of knots in the knot vector and G_i is the function value $f(y_i)$. It can be seen from eq. (B.2) that many of the basis functions will be zero. For the 0th degree basis functions (step functions), there will be only one ($p + 1$) nonzero basis function between any two knot values, indicating that only one function value contributes to the spline inside a given interval. For a cubic spline there will be four nonzero basis functions between any two knot values. This property illustrates that the B-splines are ''local'' in the sense that distant function values do not affect the spline inside a given range. It also shows that, for higher degree splines, more function values contribute to the spline within a given interval and the locality is broadened.

Equation (B.3) can be differentiated by using a recursive formula similar to eq. (B.2), which defines the derivatives of the B-spline basis func-

tions[6]:

$$\frac{d}{dx}B_p^i = \frac{p}{x_{i+p} - x_i}B_{p-1}^i(x) - \frac{p}{x_{i+p+1} - x_{i+1}}B_{p-1}^{i+1}(x). \quad \text{(B.4)}$$

To parameterize higher dimensional functions, a knot vector is required for each of the independent variables. For a 3-dimensional potential energy function, there will be three knot vectors **X**, **Y**, and **Z**. We denote the grid of sampled function values as G_{ijk}. Equation (B.3) generalizes to

$$S(x, y, z) = \sum_{i=0}^{n_x}\sum_{j=0}^{n_y}\sum_{k=0}^{n_z} B_i^p(x)B_j^p(y)B_k^p(z) \cdot G_{ijk}. \quad \text{(B.5)}$$

Equations (6) and (9) of the text follow from eq. (B.5) and from the fact that the basis functions are zero outside the range of the summation indices.

Acknowledgments

We thank J. Kostrowicki for helpful discussions and J. Y. Trosset for providing the coordinates for the thrombin/Phe-Pro-Arg system and for testing the method in his calculations. This work was supported by the National Institutes of Health (GM-14312) and the National Science Foundation (MCB95-13167). The computations were carried out on the SGI Power Challenge computers at the Cornell National Supercomputer Facility, a resource of the Center for Theory and Simulation in Science and Engineering at Cornell University, which is funded by the National Science Foundation, New York State, the IBM Corporation, and members of its Corporate Research Institute, with additional Research Resource funds from the National Institutes of Health.

References

1. P. J. Goodford, *J. Med. Chem.*, **28**, 849 (1985).
2. D. S. Goodsell and A. J. Olson, *Proteins*, **8**, 195 (1990).
3. E. C. Meng, B. K. Shoichet, and I. D. Kuntz, *J. Comput. Chem.*, **13**, 505 (1992).
4. B. A. Luty, Z. R. Wasserman, P. F. W. Stouten, C. N. Hodge, M. Zacharias, and J. A. McCammon, *J. Comput. Chem.*, **16**, 454 (1995).
5. W. H. Press, S. A. Teukolsky, W. T. Vetterling and B. P. Flannery, *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge University Press, Cambridge, U.K., Second edition, 1992, p. 123.
6. L. A. Piegl and W. Tiller, *The NURBS (Non-Uniform Rational B-Splines) Book*, Springer–Verlag, Berlin, 1995, p. 47.
7. U. Essmann, L. Perera, M. L. Berkowitz, T. Darden, H. Lee, and Lee G. Pederson, *J. Chem. Phys.*, **103**, 8577 (1995).
8. G. Némethy, K. D. Gibson, K. A. Palmer, C. N. Yoon, G. Paterlini, A. Zagari, S. Rumsey, and H. A. Scheraga, *J. Phys. Chem.*, **96**, 6472 (1992).
9. G. Hellwig, *Partial Differential Equations: An Introduction*, Blaisdell, New York, 1964, p. 36.
10. M. Minoux, *Mathematical Programming: Theory and Algorithms*, Wiley, New York, 1986, p. 102.
11. W. Bode, I. Mayr, U. Baumann, R. Huber, S. R. Stone, and J. Hofsteenge, *EMBO J.*, **8**, 3467 (1989).
12. J.-Y. Trosset, B. Maigret, and H. A. Scheraga, *J. Comput. Chem.*, to appear.